

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ "ЛЬВІВСЬКА
ПОЛІТЕХНІКА"**

Кафедра систем штучного інтелекту

Лабораторна робота № 4

з дисципліни

“Дискретна математика”

Виконав:

студент групи КН-114

Гудима Анастасія

Викладач:

Мельникова Н.І.

Львів - 2019

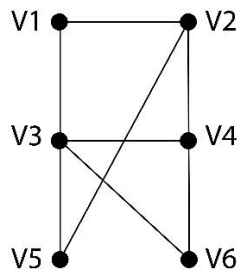
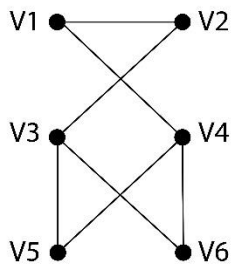
Тема. Основні операції над графами. Знаходження остова мінімальної ваги за алгоритмом Прима-Краскала.

Мета. Набуття практичних вмінь та навичок з використання алгоритмів Прима і Краскала.

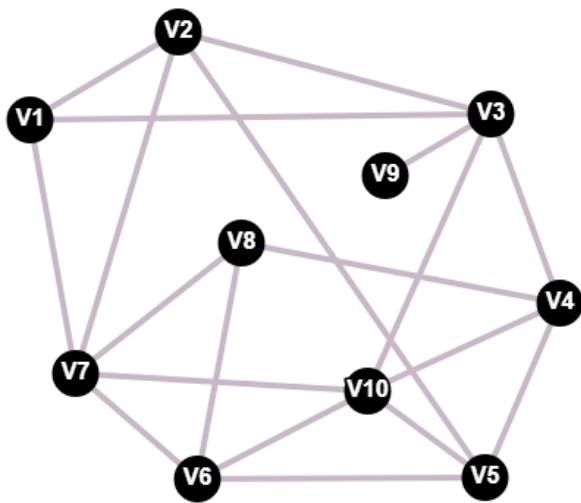
Завдання варіанту №5 з додатку 1

1. Виконати наступні операції над графами:

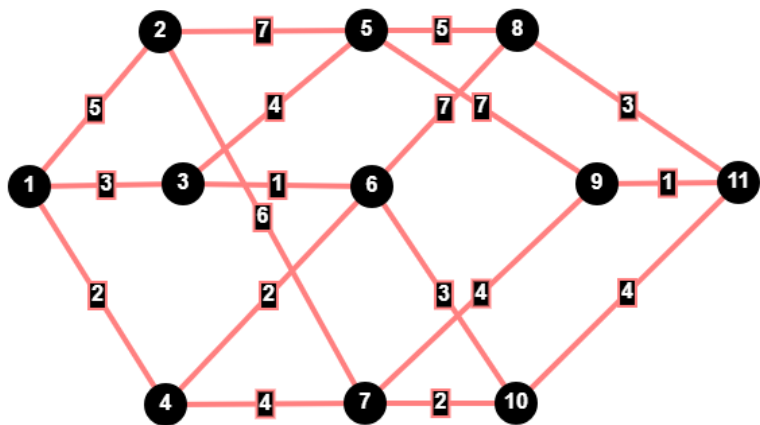
- 1) знайти доповнення до першого графу,
- 2) об'єднання графів,
- 3) кільцеву суму $G1$ та $G2$,
- 4) розщепити вершину у другому графі,
- 5) виділити підграф A , що складається з 3-х вершин в $G1$ і знайти стягнення A в $G1$ ($G1 \setminus A$),
- 6) добуток графів.



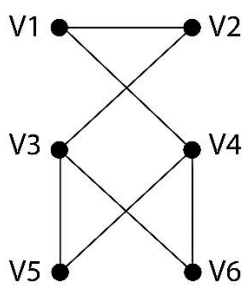
2. Знайти таблицю суміжності та діаметр графа.



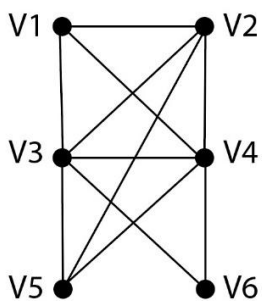
3. Знайти двома методами (Краскала і Прима) мінімальне остове дерево графа.



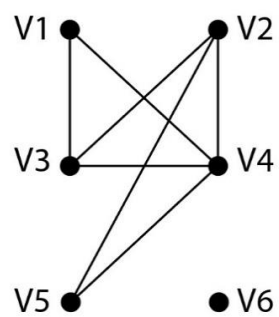
Розв'язок



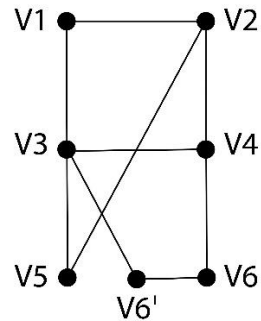
1.1)



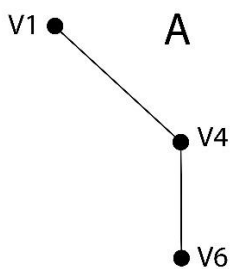
1.2)



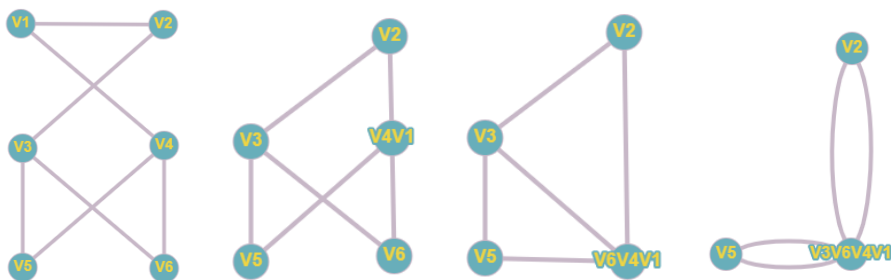
1.3)

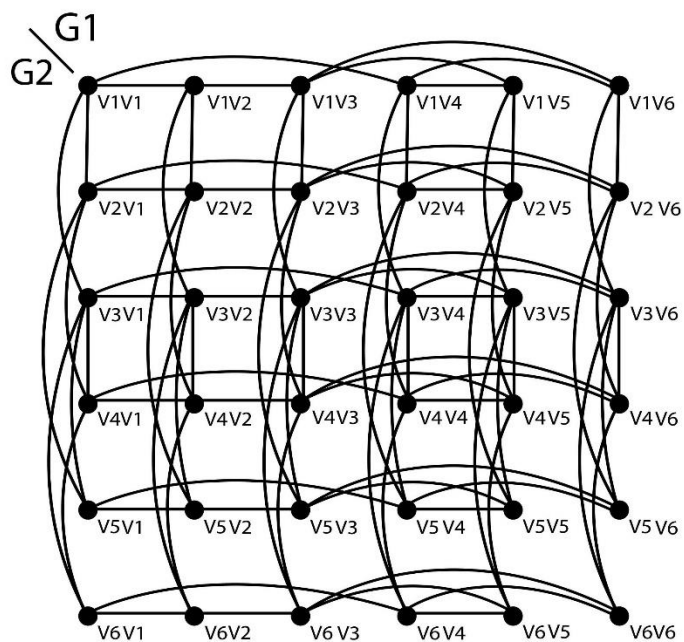


1.4)



1.5)

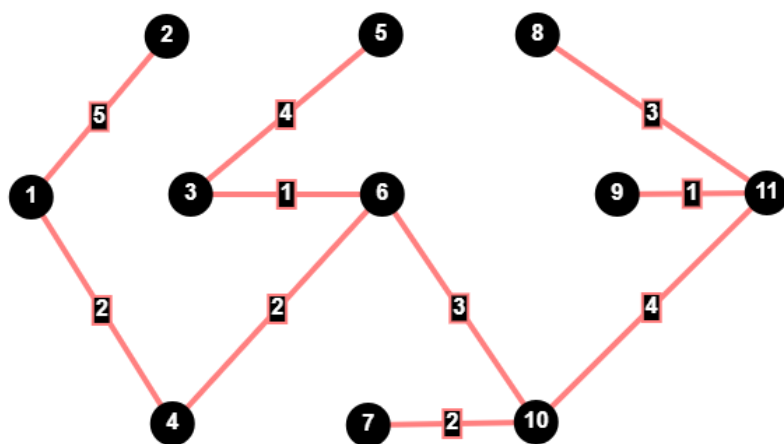




1.6)

2. Діаметр графа - 3.

| | V1 | V2 | V3 | V4 | V5 | V6 | V7 | V8 | V9 | V10 |
|-----|----|----|----|----|----|----|----|----|----|-----|
| V1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| V2 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| V3 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| V4 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| V5 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| V6 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| V7 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| V8 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| V9 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| V10 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |



3.

Алгоритм Краскала

$V = \{3, 6, 9, 11, 7, 10, 4, 1, 8, 5, 2\}$

$E = \{(3,6), (9,11), (7,10), (4,6), (1,4), (6,10), (8,11), (10,11), (3,5), (1,2)\}$

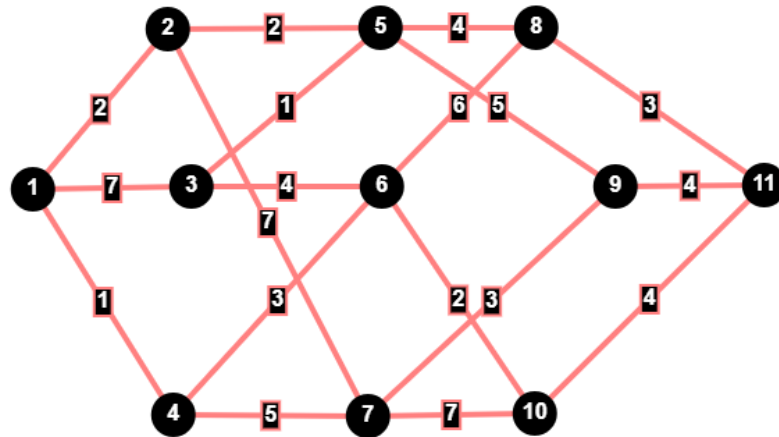
Алгоритм Прима

$V = \{6, 3, 4, 1, 10, 7, 5, 11, 9, 8, 2\}$

$E = \{(6,3), (6,4), (4,1), (6,10), (10,7), (3,5), (10,11), (11,9), (8,11), (1,2)\}$

Завдання варіанту №5 з додатку 2

За алгоритмом Прима знайти мінімальне остове дерево графа. Етапи розв'язання задачі виводити на екран. Протестувати розроблену програму на наступному графі:



Розв'язок

```
#include <iostream>

using namespace std;

struct edge
{
    int t1;
    int t2;
    int weight;
};

void vvid (edge*p, int n, int m)
{
    cout<<"Enter edges(first vertex | second vertex | weight):\n";
    for (int i=0; i<m; i++)
    {
        cout<<"Edge"<<i+1<<" : ";
        cin>>p[i].t1>>p[i].t2>>p[i].weight;
        while (p[i].t1<0 || p[i].t1>n || p[i].t2<0 || p[i].t2>n || p[i].weight<0)
        {
            cout<<"The data entered incorrectly."<<endl;
            cout<<"Try again please."<<endl;
            cin>>p[i].t1>>p[i].t2>>p[i].weight;
        }
    }
}

void bulb(edge* p, int n)
```

```

edge temp;
for (int i = 0; i < n; i++)
{
    for (int j = 0; j < n - i - 1; j++)
    {
        if (p[j].weight > p[j+1].weight)
        {
            temp = p[j];
            p[j] = p[j+1];
            p[j+1] = temp;
        }
    }
}

bool vkluchene(int* a, int n, int f)
{
    for (int i = 0; i < n; i++)
    {
        if (f == a[i])
        {
            return true;
        }
    }
    return false;
}

bool minn(int w, edge* ed, int m, int* v, int n)
{
    for (int j = 1; j < m; j++)
    {
        if (((!vkluchene(v, n, ed[j].t1) && vkluchene(v, n, ed[j].t2)) ||
            (vkluchene(v, n, ed[j].t1) && !vkluchene(v, n, ed[j].t2)))
            && ed[j].weight < w)
        {
            return false;
        }
    }
    return true;
}

void pryma (edge*ed, int*v, edge*tree, int n, int m, int&i, int&j)
{
    if (i==n)
    {return;}
    else if (j==n)
    {j=1;}
    if(vkluchene(v, n, ed[j].t1) && vkluchene(v, n, ed[j].t2))
    {
        j++;
        pryma(ed, v, tree, n, m, i, j);
    }
    else if (!vkluchene(v, n, ed[j].t1) && vkluchene(v, n, ed[j].t2)
            && minn(ed[j].weight, ed, m, v, n))
    {
        tree[i-1]=ed[j];
        v[i] = ed[j].t1;
    }
}

```

```

        j++;
        i++;
        pryma(ed, v, tree, n, m, i, j);
    }
    else if (vkluchene(v, n, ed[j].t1) && !vkluchene(v, n, ed[j].t2)
            && minn(ed[j].weight, ed, m, v, n))
    {
        tree[i-1]=ed[j];
        v[i] = ed[j].t2;

        j++;
        i++;
        pryma(ed, v, tree, n, m, i, j);
    }
    else
    {
        j++;
        pryma(ed, v, tree, n, m, i, j);
    }
}

int main()
{
    int n,m;
    cout<<"How many vertexes do you want to add? ";
    cin>>n;
    cout<<"How many edges do you want to add? ";

    cin>>m;
    cout<<endl;
    edge *ed = new edge[m];
    int *v = new int[n];
    edge *tree = new edge [n-1];
    vvid(ed,n,m);
    bulb(ed,m);
    v[0]=ed[0].t1;
    v[1]=ed[0].t2;
    tree[0]=ed[0];
    int i=2;
    int j=1;
    pryma (ed,v,tree,n,m,i,j);
    cout<<"\nV = {";
    for(int x=0;x<n;x++)
    {
        cout<<v[x]<<" ";
    }
    cout<<"\nE = { ";
    for(int x=0;x<n-1;x++)
    {
        cout<<"("<<tree[x].t1<<" "<<tree[x].t2<<" ) ";
    }
    cout<<"\n";
    return 0;
}

```

```
How many vertexes do you want to add? 11
How many edges do you want to add? 18

Enter edges(first vertex | second vertex | weight):
Edge1: 1 2 2
Edge2: 1 3 7
Edge3: 1 4 1
Edge4: 2 5 2
Edge5: 2 7 7
Edge6: 3 5 1
Edge7: 3 6 4
Edge8: 4 6 3
Edge9: 4 7 5
Edge10: 5 8 4
Edge11: 5 9 5
Edge12: 6 8 6
Edge13: 6 10 2
Edge14: 7 9 3
Edge15: 7 10 7
Edge16: 8 11 3
Edge17: 9 11 4
Edge18: 10 11 4

V = {1,4,2,5,3,6,10,8,11,9,7,}
E = { (1,4) (1,2) (2,5) (3,5) (4,6) (6,10) (5,8) (8,11) (9,11) (7,9) }

Process returned 0 (0x0)   execution time : 118.457 s
Press any key to continue.
```

Висновок: впродовж виконання лабораторної роботи я набула практичних вмінь та навичок з використання алгоритмів Прима і Краскала.